KATHOLIEKE
UNIVERSITEIT
LEUVEN

DEPARTEMENT TOEGEPASTE ECONOMISCHE WETENSCHAPPEN

RESEARCH REPORT 9954

SENSITIVITY BASED PRUNING OF INPUT VARIABLES BY MEANS OF WEIGHT CASCADED RETRAINING

by
S. VIAENE
B. BAESENS
G. DEDENE
J. VANTHIENEN
J. VANDENBULCKE

D/1999/2376/54

Sensitivity Based Pruning of Input Variables by means of Weight Cascaded Retraining

Stijn Viaene, Bart Baesens, Guido Dedene, Jan Vanthienen & Jacques Vandenbulcke

Affiliation

Leuven Institute for Research on Information Systems (LIRIS)
Department of Applied Economic Sciences
Katholieke Universiteit Leuven
Naamsestraat 69, 3000 Leuven, Belgium

E-mail

{Stijn.Viaene;Bart.Baesens;Guido.Dedene;Jan.Vanthienen;Jacques.Vandenbulcke}
@econ.kuleuven.ac.be

Acknowledgement

This work has been realised thanks to the sponsoring of and under auspices of the KBC Insurance Research Chair Management Informatics at the Department of Applied Economic Sciences of the Katholieke Universiteit Leuven. We would like to stress that the presented material is or will effectively be used in the business context of the sponsoring organization. Special thanks go out to Katrien Denys and Katrien Hulpiau for their excellent advice and practical support.

Sensitivity Based Pruning of Input Variables by means of Weight Cascaded Retraining

Stijn Viaene, Bart Baesens, Guido Dedene, Jan Vanthienen & Jacques Vandenbulcke

Leuven Institute for Research in Information Systems (LIRIS)
Department of Applied Economic Sciences
Katholieke Universiteit Leuven
Naamsestraat 69, 3000 Leuven, Belgium
{Stijn.Viaene;Bart.Baesens;Guido.Dedene;Jan.Vanthienen;Jacques.Vandenbulcke}@econ.kuleuven.ac.be

Abstract

This paper investigates the adoption of a wrapped feature selection approach using neural networks for classification purposes. The presented framework consists of a primary model selection or network construction phase and a subsequent input feature pruning phase, introduced here under the name of Weight Cascaded Retraining (WCR). The theoretical exposition in the first part of the paper will be illuminated and validated by means of real-life empirical case material. The main conclusion of the paper can be stated as follows. Feature selection can be very effective in reducing model complexity for classification modelling via neural networks. It allows one to partially circumvent the curse of dimensionality when being confronted with a high number of irrelevant/redundant features. Furthermore, by reducing the number of input features in the neural network training phase, both human understanding and computational performance can be vastly enhanced.

1 Introduction

In this paper, we are concerned with the problem of assigning members of a population to exactly one of several classes. This problematic has proven to be relevant in a wide variety of application areas ranging from marketing, over fraud detection through to health care, physics and finance. Backpropagation neural networks have shown to be very promising supervised learning tools for modelling non-linear relationships (Ripley, 1994). This, especially in situations where one is confronted with a lack of domain knowledge, which in turn prevents any valid argumentation to be made concerning model selection bias on the basis of prior knowledge. In that case, connectionist models seem a very attractive explorative choice. As universal approximators (Hornik, 1989), they can significantly improve the predictive accuracy of a classification model in comparison to linear estimation techniques. Furthermore, they too may suffer from what is often paraphrased as 'the curse of dimensionality' (Bellman, 1961) when being confronted with too many input features. Elimination of redundant and/or irrelevant features therefore often improves the predictive power of a network, in addition to reducing model complexity. Also, it need not be emphasised that models with fewer input features are capable of improving both human understanding and computational performance.

The paper investigates the use of a sensitivity based input variable pruning method introduced here as the Weight Cascaded Retraining (WCR) algorithm. This algorithm is positioned as

the second step of a two-phased wrapper approach towards feature selection using neural nets. The presented framework consists of an initial network construction (NC) phase and a subsequent WCR phase. In the first part of the paper we will outline the essentials of both phases. The basic assumption that justifies the presence of the NC phase, is that the network architecture, which is used in the initial stage of the input pruning algorithm, has to possess the inherent quality to achieve a good representation of the available data structure. In order to achieve this result, it is conceptually necessary to include a model selection phase before embarking in a subsequent variable pruning phase. The goal is then, by means of the core iteration step of the WCR algorithm, to iteratively find a neural network with 1 variable less than in the previous iteration, but without any significant degradation in generalisability. In the remainder of this paper we will outline the basic elements of the Weight Cascaded Retraining method. Focusing on the basic inter- and intra-iteration steps will provide the necessary insight into the mechanics of the suggested approach towards feature selection. In the second part of the paper, the theoretical exposition of the chosen feature selection technique will be complemented and validated by means of the application of the proposed framework to five publicly available data sets. This will function as a primary indicator of the empirical validity of the assertions that are made in the theoretical part of the exposition. Concretely, the framework will be cast upon the following publicly available real-life UCI[1] cases: the *Wisconsin Breast Cancer Database,* the *Johns Hopkins University Ionosphere Database*, the *Pima Indians Diabetes Database*, the *German Credit Approval Database* and the *Adult Database.* The set-up and semantics of each of the cases will also be briefly discussed. Results will not only provide the necessary means to benchmark the outset of this paper, they will also pinpoint the relevance of the issue of feature selection within the scope of domains as different as finance, physics and healthcare.

In more detail, the paper is constructed as follows. Section 2 provides a bird's eye view on the topic of feature selection. In Section 3, a two-stage framework for dynamic model specification will be presented. The first subsection will focus on the network construction phase, which constitutes a necessary pre-condition to the feature pruning algorithm discussed in the following subsection. The latter subsection will outline the gist of the proposed feature selection approach. Section 4 provides the necessary real-life based empirical evidence illustrating the assertions discussed in the foregoing section. The discussion is concluded with a final section summarising the main findings and some remarks regarding future research.

2 A Bird's Eye View on Feature Selection

This section will enable the reader to situate the presented effort amidst the vast multitude of feature extraction methods that have been proposed in the literature on neural networks. As feature selection is a kind of feature extraction, it is appropriate to briefly zoom out to this broader context, before zooming in to the realm of feature selection methods.

Feature extraction is to be situated as a means of dimension reduction (Carreira-Perpiñán, 1997). The goal hereof is to find a new and reduced co-ordinate system that allows to project the data samples on a more compact representation, allowing to reduce model complexity.

---

[1] University of California Irvine Repository available at: http://kdd.ics.uci.edu/

More formally, the problem can be considered as finding the optimal mapping φ from a p-dimensional input space to a q-dimensional space, with $q < p$, such that some performance criterion is optimised (e.g. minimisation of information loss). The general assumption underlying this operation and justifying it, is that the data sample approximately lies within the bounds of this reduced space. Often the representation of the available data will in fact be redundant. Redundancy can be induced by irrelevant features, i.e. features that show a variation equivalent to a noise factor. Another common redundancy factor relies in the presence of correlation among features. To further appreciate the beneficial effect of dimension reduction in the context of supervised learning, one immediately encounters the phenomenon termed by Bellman as 'the curse of dimensionality' (Bellman, 1961). It refers to the fact that, in the absence of simplifying assumptions, the sample size needed to estimate a function of several variables to a given degree of accuracy, grows exponentially with the number of variables. High dimensional spaces are often inherently sparse, essentially avoiding the algorithm from effectively and efficiently generalising out of the available data points. Hence, feature extraction methods can play a pivotal role in any knowledge discovery process. Now, zooming back in, the focus of this paper lies on feature *selection*, a technique towards feature extraction that binarily filters a set of given input features describing the data samples at hand for training a connectionist model (John, 1994). Thus, although undoubtedly potentially very interesting, methods based on Principal Component Analysis (Jolliffe, 1972) and other inherent feature *construction* methods (Piramuthu, 1998) will not reside within the scope of this exposition.

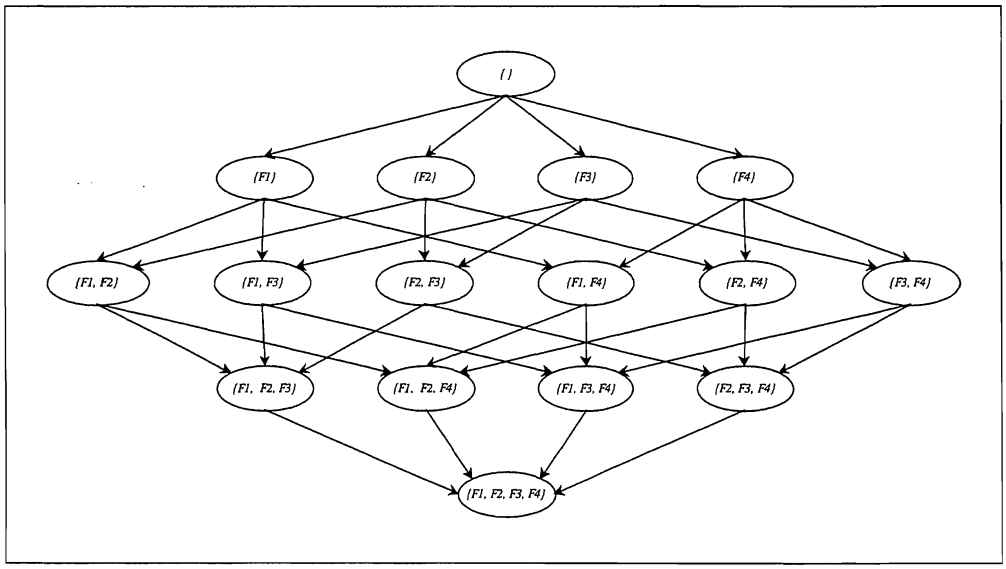

Figure 1: Feature Search Space Representation

An optimal feature subset can only be obtained when the feature space is exhaustively searched. Figure 1 shows a common representation of the search space. When k features are present, this would imply the need to evaluate 2^k-1 feature subsets. Unfortunately, as k grows, this very quickly becomes computationally infeasible. For that reason, a heuristic search procedure through the vast search space is often preferred.

Essentially, the inception of a feature space search algorithm boils down to a choice of several search-guiding parameters. These heuristics can be typified by means of the generic choices concerning the parameterisation of their state search procedure. Major variation points include (Langley, 1994):

- the choice of an appropriate *starting point* within the search space, where two obvious alternatives are bottom up and top down, i.e. respectively starting from the empty and the full feature set. Any starting point in between could also do the trick, but most methods implement one of the two above.

- the choice of an appropriate *search heuristic*, covering the criterion to iteratively approximate a candidate solution. Here, amongst a multitude of alternatives, the best first heuristic is used extensively.

- the choice of a *goal function*, to be optimised over subsequent steps of the algorithm at hand. While the search strategy proposes candidate feature subsets, the goal function decides whether one is superior to the others.

Feature Selection can either be performed as a pre-processing step to the actual learning algorithm or be completely integrated herein. The former approach is termed 'filter' , the latter 'wrapper' (Kohavi, 1996). Two well-known filter approaches include FOCUS (Almuallim, 1991) and Relief (Kira, 1992). In the wrapper approach, feature selection is integrated within the learning steps of the induction algorithm itself. The latter then, iteratively, provides feedback about a pre-specified performance characteristic of the presented feature subset. As a result, features may be added and/or removed from consideration into the next iteration of the learning algorithm, until, eventually, some desired feature subset is obtained. The C4.5 decision tree algorithm (Quinlan, 1993) is an excellent illustration of the mechanics of this process. The WCR algorithm, as introduced in the next section of this document, is implemented as a typical wrapper approach towards feature selection, using a (greedy) best-first heuristic to guide the backward search procedure within the feature set solution space.

3 A Framework Towards Dynamic Model Specification

The framework to be presented consists of a primary model selection or network construction (NC) phase and a subsequent Weight Cascaded Retraining (WCR) phase. In this section, we will outline both phases in detail. Figure 2 gives an outline of the essential inputs and outputs to be expected from either phase within the set-up. The eventual output of the procedure is expected to be a triplet {A*,F*,W*}, in which A*, F* and W* respectively stand for the final architecture of the connectionist model, i.e. the number of hidden layers, neurons and connections, F* stands for the optimally reduced feature set, given A*, and W* stands for the optimised weight set of A* trained with input set F*.

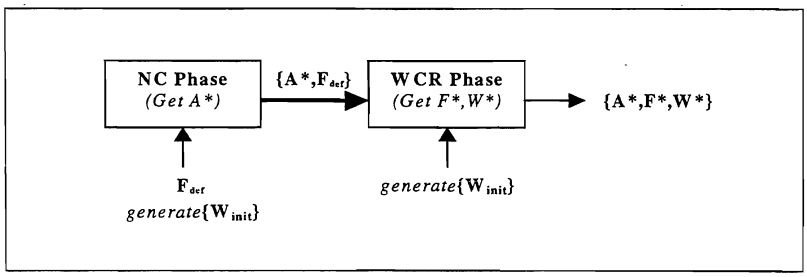

Figure 2: A Two-Step Dynamic Model Specification Approach

The basic assumption that underlies this sequential procedure is that the network architecture A*, which is used in the initial phase of the input pruning algorithm, i.e. at the start of the first algorithmic WCR iteration, has the inherent quality to achieve a good representation of the available data structure (Van De Laar, 1999). In order to achieve this, it is conceptually advisable to include a model selection phase before embarking in any variable pruning phase. The NC step takes the total set of available inputs F_{def} as a given. The goal is then, in a second phase, iteratively, by means of the basic iteration step of the WCR algorithm outlined in section 3.2, to find a neural network with one variable less, but without any significant degradation in generalisability. In addition, since the performance of any connectionist model is known to be highly dependent upon the choice of the initial weight set W_{init}, it is strongly suggested to integrate a W_{init} generation function into both of the sequential/dynamic network construction steps. Re-sampling of initial weights is claimed to be a necessary pre-condition in order to be able to formulate statistically relevant assertions within a context of evaluating neural network performance (Moody, 1992 ; Refenes, 1999).

In the remainder of this section we will outline the basic elements of the Weight Cascaded Retraining method, after briefly having discussed the major options that have been brought forward in the literature on network construction. Focusing on the core inter- and intra-iteration steps will provide the necessary insight in the dynamics of the approach.

3.1 A Taste of Network Construction (NC)

For most real-life problems, the optimal architecture of the connectionist model to be conceived is not known in advance. Regretfully, there is no structured optimal means to determine the number of hidden layers and the number of processing units in these layers. Still, sizing the neural network remains a consideration of the utmost importance (Wang, 1994 ; Murata, 1994 ; Moody, 1991). A neural network solution having too small an architecture to capture the underlying functional relationships, will not achieve satisfactory predictive accuracy, let alone generalisability. A neural network with too large an architecture will not only fit the effective relationships inherent to the whole instance population, it will even try to fit the idiosyncrasies of the training set. This overfitting of the training data, which essentially boils down to memorising noise, will generally lead to very good training performance but rather poor test set performance.

The above considerations have often resulted in pragmatic architecture selection by means of trial and error or by means of empirically obtained heuristics[2]. There are literally tons of approximate architecture selection methods available for integration into the NC phase (Ripley, 1995). A first level categorisation of methods can be made on the basis of the applied search strategy in architecture space. Two straightforward sequential approaches are forward and backward selection, known to the neural network society as growing and pruning strategies. Whereas a growing strategy incrementally adds hidden neurons to the candidate solution while starting from the empty model, pruning strategies do just the opposite, starting from a pre-determined maximal architecture. Examples include the stepwise model construction approach known as the Cascade Correlation learning algorithm (Fahlman,1990), the SNC approach described in Moody (Moody, 1992), or the algorithm suggested in (Setiono, 1996). Besides pruning neurons, there are also methods that focus on the removal of individual connections. In (Reed, 1993), a tentative overview of pruning algorithms is given. Among the alternatives we identify the very popular OBS (Hassibi, 1993) and OBD (Le Cun, 1990) techniques.

A second element, which differentiates NC implementations, is their model selection criterion. When having obtained an initial insight into the performance of a whole range of candidate architectures, it has to be possible to choose one for further fine-tuning within a subsequent phase, in our case the WCR phase. Many different criteria can be conceived. For instance, there are those based on the generalising capabilities of the candidate solution as measured by the performance on an independent test set or alternatively by some error measure based on training set performance, taking into account possible correction factors for overfitting on the training data (Le Cun, 1990 ; Hassibi, 1993 ; Moody,1991). Alternative measures include, among many others, the use of a Network Information Criterion (Murata, 1994) and of the principle of Minimum Description Length (Rissanen, 1978).

At this moment, there is no general agreement on which of these methods or which combination hereof is superior. Systematic experimental evaluation and characterisation of the proposed methods is needed in order to get a clearer picture of the effects of each of these approaches conditional upon the circumstances.

---

[2] Neural-Works (1996), for instance, recommends (number of inputs + outputs)*2/3 as the number of hidden units using only one hidden layer, though they also recommend thorough experimentation by increasing/decreasing the number of units.

3.2 The Weight Cascaded Retraining (WCR) Algorithm

The approach implemented can best be typified as a typical wrapper-approach (John, 1994) using a best-first search heuristic to guide our backward search procedure towards the optimal feature set. The skeleton of the algorithm is depicted in Figure 3.

Initialisation phase

$F_0 = \{f_1, \ldots, f_n\}$; // initial set of n features

construct initial network;

initialize network weights;

Core iteration

for i=0 to $|F_0|$ do

begin

train network;

for j=1 to $|F_i|$

begin

$\theta(f_j) = \Gamma(F_i) - \Gamma(F_i | f_j = \text{mean}(f_j))$;

// compute sensitivity index for all features in F_i

end

$f_p = \text{argmin}_{f_k \in F_i} \theta(f_k)$;

// detect feature with lowest θ

$F_{i+1} = \{f_j \in F_i \mid \theta(f_j) > \theta(f_p)\}$

prune f_p and remove its connections with the first hidden layer;

end

Figure 3: The WCR Algorithm Outlined

In an initialisation phase to the core iteration of the WCR algorithm, the initial network architecture A* is constructed. This is done according to the parametric choices made in the NC phase, that precedes the feature pruning phase outlined here. By default, we start the procedure in a top down fashion, i.e. with a full feature set $F_0 = \{f_1, \ldots, f_n\}$. This implies the creation of an input layer with $|F_0|$ input neurons. The description of A* will furthermore contain the specification of the number of hidden units to be used, as well as the number of hidden layers and connections. The number of output neurons usually is pre-determined by the encodings of the problem data and therefore is treated as a given not to be optimised in the NC phase. The weights and biases on the relevant connections between neurons of the network are then initialised. In the implemented version of the algorithm we opted for weight initialisation according to the Nguyen-Widrow algorithm. Instead of choosing purely random

values for the set of starting weights, this rule initialises the relevant connections more favourably, which allows to improve the generalisability and to speed up the function approximation (Nguyen, 1990).

Starting with all inputs, the core of the algorithm – *termed as 'Core iteration' in Figure 3* – will run $|F_0| + 1$ times, each time training and evaluating a network in order to delete the least significant variable from within the remaining feature set F_i. All inputs are pruned sequentially, i.c. one by one. Thus, each core iteration step starts with training a reduced network with an appropriately reduced set of inputs. Notice that the last network to be evaluated, by default, will have no inputs. For a classification problem, the final network will be equivalent to a purely random model, classifying all instances of the training set in one of the classes.

The default choice neural network training algorithm is standard backpropagation. It may be advisable to use an adaptive learning rate. In that way, the learning step size will dynamically take on smaller values as the training algorithm proceeds. In addition, a momentum term may be added to avoid getting stuck in local minima. Advanced training algorithms (e.g. Levenberg-Marquardt, Quasi-Newton and conjugate gradient methods) might be used to speed up the convergence.

The inner iteration computes a sensitivity measure $\theta(f_j)$ for each of the $|F_i|$ remaining input features. This sensitivity metric essentially assesses the importance of the input feature vis-a-vis the network's performance. There is no unique quantification of an input feature's importance (Refenes, 1999). Note that the concept of sensitivity of the model to the presence/absence of a feature, as defined by the above sensitivity measure, does not completely correspond to the concept of causal relevance of a feature within the real, but unknown, functional relationship. Interaction and correlation effects among features tend to obscure a rightful assessment of the causal relevance of a feature.

We propose to define the sensitivity index $\theta(f_j)$ of a feature f_j contained within the current feature set F_i as follows :

$$\theta(f_j)=\Gamma(F_i)-\Gamma(F_i|\, f_j = \text{mean}(f_j))$$

Following Moody et al. (Moody, 1992), we perturb each input feature to its mean and compute the impact on the network output by means of an error measure Γ (typically a mean squared error MSE). This amounts to a strategy of constant substitution, treating the input to be neglected as missing by substituting its effect to its mean over the whole sample. Notice that no retraining is needed while computing these sensitivities.

The suggested approach proves to be fairly robust with respect to interaction and correlation. As to the presence of interaction effects, suppose two variables are interacting in a significant fashion. Setting one variable to its mean will destroy the interaction and consequently degrade the network performance, resulting in a large value for the sensitivity index $\theta(fj)$. As to the presence of correlation effects, consider for instance the extreme situation in which two variables are perfectly correlated and at least one of them has an inherent significant causal contribution within the functional relationship to be learned. At first sight, the network will seem individually insensitive to either one of these variables, since no information is lost by holding either one constant, leading to the pruning of one of them. However, after having removed the first one, the relevance of the other feature increases dramatically. For that

reason, re-computing the sensitivity indices in the next iterative step will provide evidence on the significance of the remaining feature. So, since the relevance of a variable may change upon removal of another variable, the sensitivity of all remaining features will be recomputed in each core iteration step.

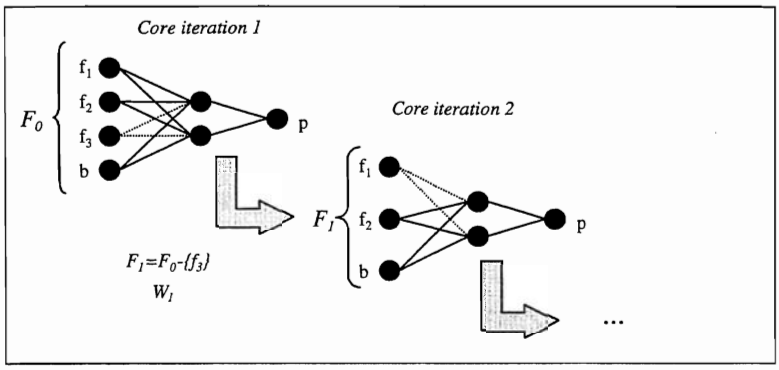

Figure 4: Illustration of the Cascading Nature of WCR

Resuming: at the end of each core iteration step the procedure will greedily remove the input node belonging to the feature with the lowest sensitivity index among the remaining features. As a direct corrollary of this operation, all its neural connections with the first hidden layer will be omitted from further consideration. At this point, the weight values of all other connections remain unchanged. By passing the optimised weight set W_i to the next iteration, we try to provide the network with a better starting point and check whether the network can perform better with one feature less. There exist other methods in the literature on neural nets that appeal to the same idea of starting with an improved set of weights (Van De Laar, 1999 ; Egmont-Petersen, 1998). The mechanics and cascading nature of the core of the WCR algorithm are visually illustrated in Figure 4.

4 Empirical Validation

In the previous part of the paper, we proposed a two-phased approach to dynamic connectionist model selection. The gist of the approach amounts to the application of the WCR-based feature pruning algorithm to a candidate connectionist model output from the NC phase. From hereon, we report the experimental results of applying the previously presented framework to five publicly available benchmark data sets. These standard benchmarks will function as primary indicators of the validity of the assertions that are made in the theoretical part of the exposition. This will concretely be illustrated by means of the following data sets: the *Wisconsin Breast Cancer Database,* the *Johns Hopkins University Ionosphere Database,* the *Pima Indians Diabetes Database*, the *German Credit Approval Database* and the *Adult Database*. Data for all of the problems used as benchmark in this report can be obtained via anonymous ftp from the University of California-Irvine (UCI) Repository. Specific domain theoretic considerations will be kept to a minimum. Details on, as well as past usage of the cases can be obtained via the referenced web site. Results will not only provide the necessary means to benchmark the outset of this paper, they will also pinpoint the relevance of the issue

of feature selection within the scope of domains ranging from health care, over physics to finance.

As a pre-processing stage to the learning algorithm, every feature was statistically normalised to a mean of 0 and a standard deviation of 1. In the case of missing feature values, records were simply eliminated from consideration.

We opted for one hidden layer in all of the cases, influenced by theoretical works, which show that a single hidden layer is sufficient to approximate any complex non-linear function with any desired degree of accuracy (Hornik, 1989). Furthermore, we used logistic activation functions for all hidden and output nodes. As for the implementation of the training algorithm, the choice was consistent over all five cases: backpropagation neural learning with momentum and adaptive learning rate. In order to eliminate the randomness introduced by the initial weight set choice, the WCR algorithm was run several times, starting from a different W_{init} set, making it possible to assess the results on statistical grounds.

4.1 Wisconsin Breast Cancer Database

This breast cancer database was obtained from the University of Wisconsin Hospitals, Madison. It is a data set consisting of 699 records of which 458 represent benign samples and 241 malignant. Each record consists of 9 meaningful attributes, i.c. measurements taken from a fine needle aspirates from a patient's breast. Feature valuation was done at the time of the assessment on a scale from 1 to 10, with 1 being the closest to benign and 10 being the most anaplastic. On the whole database, there were 16 records with missing values. They were deleted from the processing set. For our purposes, the 683 remaining instances of the database were randomly assigned to one of three sets: a training set (315), a validation set (36) for early stopping and an independent test set (332).

Table 1 summarises the main architectural and algorithmic choices for this problem. In the network construction step, we experimented with several fully connected feed-forward networks, varying the number of hidden units within the range of [12:20]. The final choice architecture A* amounts to the best generalising network, taking into account the trade-off between model complexity and model accuracy.

Architecture and Algorithm	*Wisconsin*
Number of input neurons ($\in A^*$)	9
Number of hidden layers ($\in A^*$)	1
Number of hidden neurons ($\in A^*$)	12
Number of output neurons ($\in A^*$)	1
Number of epochs	1000
Training set size	315
Validation set size	36
Test set size	332

Table 1: Implementation Choices for Wisconsin Database

The following figure gives an indication of the performance of the WCR pruning algorithm. As the algorithm proceeds through the steps of its core iteration, we plot the MSE and PCC[3] curves averaged over 50 well generalising runs of the WCR algorithm. At the start of each of these runs, the network weights W_{init} were randomly initialised according to the Nguyen-Widrow rule. Both training and test set performance are depicted.

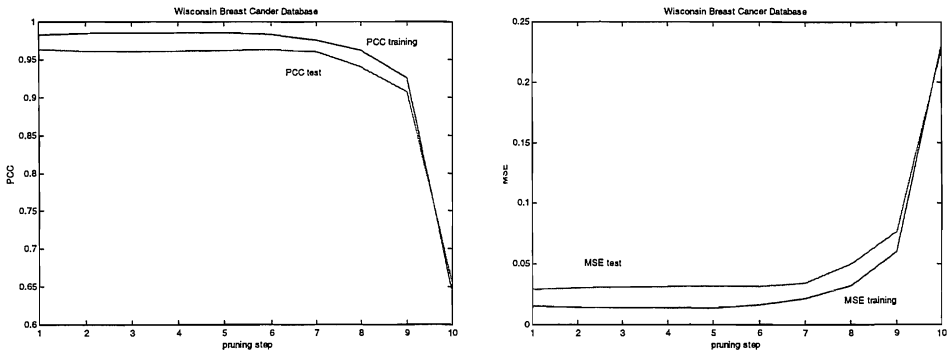

Figure 5: PCC and MSE Curves for Wisconsin Database

Looking at Figure 5, the question naturally arises where to situate the optimal cut-off feature set. From the point of view of feature extraction, one has to give a robust indication of the size of the preferred final feature set. Visual methods would be a straightforward choice. However, statistically based methods are to be preferred. Furthermore, a trade-off needs to be made between model complexity and model accuracy, a.k.a. the bias/variance trade-off (Friedman, 1997 ; Geman, 1992). Several criteria have been devised to effectively cope with this trade-off, e.g. Network Information Criterion (Murata, 1994), Akaike Information Criterion (Akaike, 1974). In this paper, we will determine the cut-off point by means of an Analysis of Variance (ANOVA) approach.

The procedure is fairly straightforward. First, we start by identifying the top of the mean PCC curve on the training set. Naïve reasoning would then go for the feature set at this point as the optimal cut-off. For the Wisconsin data, this would lay the cut-off at pruning step 4. The resulting feature set would then consist of 6 features. However, the cut-off decision would then be purely based on a mean performance criterion evaluated over the training set. In order to take into account the beneficial effect of reduced model complexity (cf. bias/variance trade-off), we proceed with a sequence of ANOVA tests. In subsequent steps, we proceed along the mean PCC_{train} curve, starting at pruning step 4 (i.e. maximum mean PCC_{train}) and perform a one-way ANOVA analysis to determine the point at which the mean PCC_{train} value decreases significantly (5% significance level). This procedure allows to take into account the variance of the PCC_{train} values over all 50 well generalising runs of the WCR algorithm. The process is illustrated in Figure 6.

[3] Percentage Correctly Classified

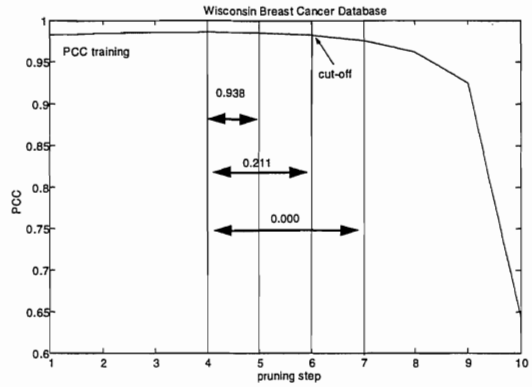

Figure 6: Illustrating the ANOVA Approach

The number above each horizontal arrow in Figure 6 represents the p-value of the corresponding ANOVA test. In concreto, the ANOVA on the mean PCC_{train} values applied over position ranges [4:5] and [4:6] respectively returns a p-value of 0,938 and 0,211, thereby accepting the null hypothesis that the mean PCC_{train} values within these ranges do not significantly differ. This is equivalent to accepting a null hypothesis stating it is statistically justified to fit a horizontal line through this limited section of the curve. However, trying to fit a horizontal line through the Mean PCC_{train} curve over positions 4 to 7 is statistically rejected, meaning that the mean PCC_{train} value at position 7 significantly differs from those in positions 4, 5 and 6. This reasoning process naturally leads to the identification of the cut-off point at position 6, as illustrated in Figure 6. So, at the end of this process, we opt for a model with a feature set F_m* consisting of 4 remaining features as our best choice. This choice and some other indicative performance metrics are summarised in the column of Table 2 labelled F_m*. The results in this column can be contrasted with the results in the column labelled F_s*. In the latter, the optimal number of features was determined using a criterion very similar to that used by Setiono et al. in the feature selection algorithm that was proposed in (Setiono, 1997). Basically, this comes down to identifying a cut-off position for each of the 50 WCR runs separately, based upon their maximum PCC_{train} value. Averaging over this set of cut-off positions yields the resulting number of remaining features (5.33).

Mean Results	F_o	F_m*	F_s*
MSE_{train}	0.015	0.016	0.015
PCC_{train}	0.983	0.983	0.983
MAD_{train}	0.040	0.031	0.033
MSE_{test}	0.029	0.031	0.031
PCC_{test}	0.963	0.963	0.961
MAD_{test}	0.056	0.047	0.050
# features	9	4	5.33

Table 2: Performance Metrics for the Wisconsin Database

Notice that the suggested algorithm at this stage is only partially deterministic, in that it provides an indication of the number of important features, however, without identifying which ones. This as a direct result of the fact that the order in which features are eliminated in a neural network feature selection method may be dependent upon randomly initialised network parameters, e.g. W_{init}. Thus, while each run of the algorithm does rank features in ascending order of discriminative power, it is advisable not to attach too much significance to the ranking of features over a single run of the WCR algorithm. It is far better to look at trends across a series of experiments. A straightforward way of doing this is to rank the features according to their mean pruning position over the 50 runs.

In sections 4.2 to 4.5, the above generic procedure will be cast upon four other publicly available data sets. Each of the cases will be briefly described, while results for all cases will be reported by means of the included figures and tables. The findings that are reported in section 4.5 are especially interesting as the discussed approach is evaluated on one of the larger UCI data sets.

4.2 Johns Hopkins University Ionosphere Database

This radar data was collected by a system in Goose Bay, Labrador. The system consists of a phased array of 16 high-frequency antennas with a total transmitted power on the order of 6.4 kilowatts. The targets were free electrons in the ionosphere. "Good" radar returns are those showing evidence of some type of structure in the ionosphere. "Bad" returns are those that do not. The database consists of 351 samples, each covering 34 continuous attribute values and 1 binary classification attribute. There were no missing values within the data set. For our purposes, the 351 instances of the database were randomly assigned to one of three sets: a training set (175), a validation set (26) for early stopping and an independent test set (150).

4.3 Pima Indians Diabetes Database

The data set contains 768 records of female Pima Indians, which may show signs of diabetes. Each record is made up of 8 attributes partially describing the patient's medical history. Among the features are elements as diastolic blood pressure, number of times pregnant and age, which are presumed to be relevant indicators to predict whether the patient tested positive for diabetes. This data set has been referred to numerous times in the literature. It is considered quite a challenge on which even state of the art neural techniques still misclassify about one fourth of the population. For our purposes, the 768 instances of the database were randomly assigned to one of three sets: a training set (345), a validation set (51) for early stopping and an independent test set (302).

4.4 German Credit Approval Database

This database is also part of the STATLOG project database. It contains 1000 records, each of them described by 20 attributes. No missing values were reported. For our purposes, the 1000 instances of the database were randomly assigned to one of three sets: a training set (500), a validation set (100) for early stopping and an independent test set (400).

4.5 Adult Database

To illustrate the potential of the proposed methodology for data mining purposes, it was also applied to the much larger publicly available Adult UCI database. The Adult database involves the prediction whether income exceeds $50K/yr based on census data. It consists of 45122 observations, each having 14 attributes. The implementation choices and the results of the application of the WCR algorithm to the Adult Database are summarised in Figure 10 and Tables 3 and 4.

For what it's worth, in order to give some indication as to the time overhead incurred by the proposed algorithm, it took the WCR phase on average about 20 minutes to process the Adult data set on a Pentium III 450 MHz processor with 128 Mb Ram running Windows NT Server 4.0. What is more important is that we also ran simulations in order to assess the speed-up in convergence realised by the weight passing from one iteration to the next. We contrasted the discussed WCR approach with a cascading set-up without the element of weight passing, i.e. initialising the network in each pruning step with a weight vector set by the Nguyen-Widrow rule. Over the fifty runs of the WCR algorithm, we noticed a consistent speed-up in convergence of the algorithm of over 50% for the Adult data set due to the effect of passing the optimised set of weights.

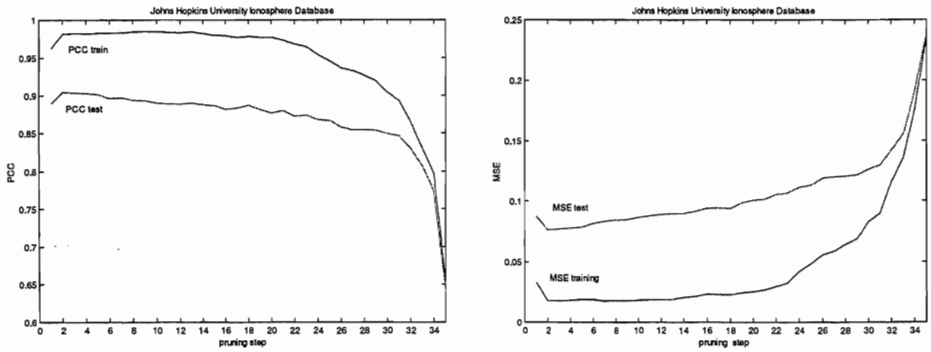

Figure 7: PCC and MSE Curves for Ionosphere Database

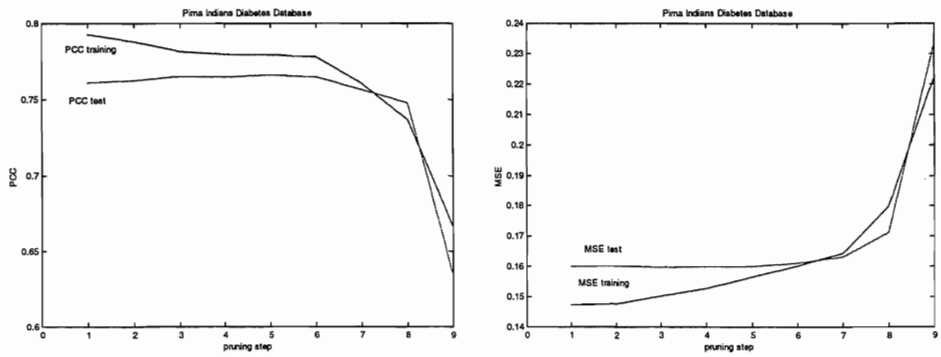

Figure 8: PCC and MSE Curves for Pima Indians Database

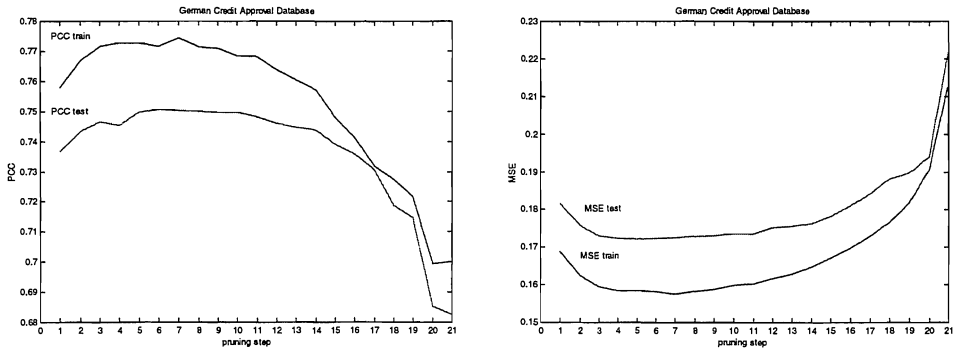

Figure 9: PCC and MSE Curves for German Credit Database

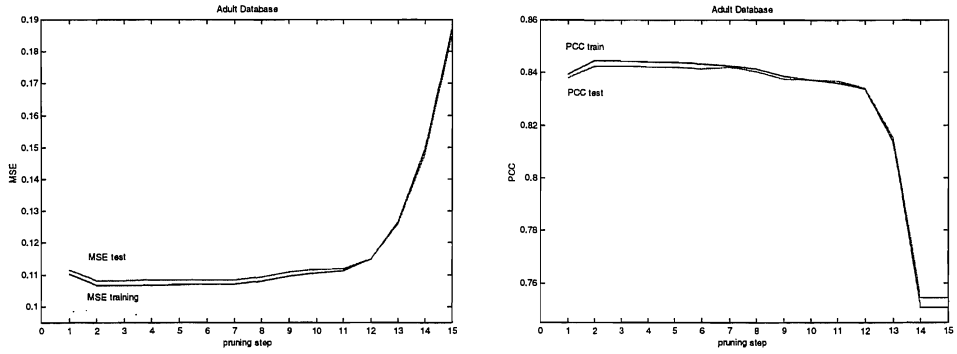

Figure 10: PCC and MSE Curves for Adult Database

Architecture and Algorithm	Iono	Pima	German	Adult
Number of input neurons	34	8	20	14
Number of hidden layers	1	1	1	1
Number of hidden neurons	16	12	14	6
Number of output neurons	1	1	1	1
Number of epochs	1000	1000	1000	1000
Training set size	175	345	500	25499
Validation set size	26	40	100	4663
Test set size	150	383	400	15060

Table 3: Implementation Choices for Public Data Sets

Mean Results	F_o				F^*			
	Iono	**Pima**	**German**	**Adult**	**Iono**	**Pima**	**German**	**Adult**
MSE_{train}	0.032	0.147	0.169	0.110	0.026	0.148	0.162	0.107
PCC_{train}	0.963	0.793	0.758	0.839	0.973	0.788	0.760	0.844
MAD_{train}	0.098	0.303	0.324	0.220	0.082	0.302	0.320	0.216
MSE_{test}	0.087	0.160	0.181	0.111	0.101	0.160	0.175	0.108
PCC_{test}	0.890	0.761	0.737	0.838	0.880	0.762	0.745	0.842
MAD_{test}	0.168	0.319	0.330	0.221	0.161	0.318	0.323	0.218
# features	34	8	20	14	14	7	8	10

Table 4: Obtained Results for Public Data Sets

5 Conclusion

The paper investigates the use of a sensitivity based input variable pruning method introduced here as the Weight Cascaded Retraining (WCR) algorithm. This algorithm is positioned as the second step of a two-phased wrapper approach towards feature selection using neural nets. A bird's eye view on feature selection is presented along with the theoretical exposition of the proposed framework. Transposing the presented framework onto five publicly available benchmark data sets yields the necessary experimental evidence. The main conclusion of the paper can be stated as follows. Feature selection can be very effective in reducing model complexity for classification purposes via neural networks. It allows one to partially circumvent the curse of dimensionality when being confronted with a high number of irrelevant/redundant features. Furthermore, by reducing the number of input features in the neural network training phase, both human understanding and computational performance can be vastly enhanced. However, all but the last word has been said with respect to this research topic. Topics of further research include:

- the investigation of the interaction between the NC and the WCR phase. Here both phases have been presented as purely sequential in nature. An iterative approach may further enhance the reported results (Moody, 1992).

- the sensitivity of the algorithm to the chosen feature selection criterion. Although we have only presented one kind of sensitivity measure in this paper, others may prove to be useful.

- an extensive and thorough comparison of the proposed feature selection approach with other in the literature established approaches. For a thorough overview of some of the main feature selection methods used for connectionist modelling, we refer to (Bonnlander, 1996).

Furthermore, while the algorithm only briefly discusses the topic of identifying the significant features, this remains a very interesting as well as challenging topic for further discussion. As stated in the text, relevance of a feature is not equivalent to significance. Interaction and correlation effects may play a misguiding role in any assessment of a feature's relevance. It should also be clear from the above discussion, that the outlined WCR algorithm inherently holds a great amount of potential to optimise neural network performance and comprehension. However, the dependence of the WCR algorithm on the outcome of the NC phase needs further investigation. Therefore, extensive experimentation with alternative NC

algorithms will be undertaken, in order to make optimal use of the rationale embedded in the WCR algorithm.

6 References

Akaike H., A new look at statistical model identification, *IEEE Transaction on Automatic Control*, 19, 1974, pp. 716-723.

Almuallim H. and Dietterich T.G., Learning with many irrelevant features, *Proceedings of the 9th National Conference on Artificial Intelligence*, 1991, pp. 547-552.

Bellman R.,. Adaptive Control Processes: A Guided Tour, Princeton, University Press,1961.

Bonnlander B., Non-parametric Selection of Input Variables for Connectionist Learning, Phd. thesis, Department of Computer Science, University of Colorado, 1996.

Carreira-Perpiñán .A., A Review of Dimension Reduction Techniques, Technical Report CS-96-09, Department of Computer Science, University of Sheffield, 1997.

Egmont-Petersen M., Talmon J.L., Hasman A. and Ambergen A. W., Assessing the Importance of Features for Multi-Layer Perceptrons, *Neural Networks*, 11, 1998, pp.623-635.

Fahlman S.E. and Lebiere C., The cascade-correlation learning architecture, *NIPS2*, 1990, pp. 524-532.

Friedman J., On Bias, Variance, 0/1-Loss, and the Curse of Dimensionality, *Data Mining and Knowledge Discovery 1*, 1997, pp.55-77.

Geman S., Bienenstock E. and Doursat R., Neural Networks and the bias/variance dilemma, *Neural Computation*, 4, 1992, pp. 1-58.

Hassibi B., Stork D.and Wolff G., Optimal Brain Surgeon and General Network Pruning, *Proceedings of the IEEE International Conference on Neural Networks*, San Francisco, Vol. 1, 1993,.pp.293-299.

Hornik K., Stinchcombe M. and White H., Multilayer Feedforward Networks are Universal Approximators, *Neural Networks*, 2, 1989, pp.359-366.

John G.W., Kohavi R. and Pfleger K., Irrelevant Features and the Subset Selection Problem, *Machine Learning: Proceedings of te Eleventh International Conference*, Morgan Kaufmann Publishers, San Francisco CA, 1994, pp. 121-129.

Jolliffe I.T., Principal Component Analysis, NewYork: Wiley, 1972.

Kira K., and Rendell L.A., The feature selection problem : Traditional methods and a new algorithm, *Proceedings of the 10th National Conference on Artificial Intelligence,* 1992, pp. 129-134.

Kohavi R. and John G, Wrappers for Feature Subset Selection, *Artificial Intelligence journal, special issue on relevance*, Vol. 97, No 1-2, 1996, pp. 273-324.

Langley P., Selection of Relevant Features in Machine Learning, *AAAI Fall Symposium on Relevance*, 1994, pp. 140-144.

Le Cun Y. , Denker J.S.and Solla S.A., Optimal Brain Damage, *Proceedings of the Neural Information Processing Systems*, 2, D.S. Touretzky (ed.), Morgan Kaufmann, 1990, pp. 598-605.

Moody J.and Utans J., Principled Architecture Selection for Neural Networks: Application to Corporate Bond Rating Prediction, *NIPS4*, 1992, pp. 683-690.

Moody J.E., Note on Generalization, Regularization and Architecture Selection in Nonlinear Learning Systems, *First IEEE-SP Workshop on Neural Networks for Signal Processing*, IEEE Computer Society Press, Los Alamitos, CA, 1991, pp. 1-10.

Murata N., Yoshizawa S.and Amari A., Network Information Criterion - Determining the Number of Hidden Units for an Artificial Neural Network Model, *IEEE Transactions on Neural Networks*, 5, 1994, pp. 865-872.

Nguyen D. and Widrow B., Neural Networks for Self-Learning Control Systems, *IEEE Control Systems Magazine*, 1990, pp. 18-23.

Piramuthu S., Ragavan H. and Shaw M.J., Using Feature Construction to Improve the Performance of Neural Networks, *Management Science,* Vol. 44, No. 3, 1998.

Reed R., Pruning Algorithms - A Survey, *IEEE Transactions on Neural Networks*, Vol. 4, No. 5, September 1993, pp. 740-747.

Quinlan J.R., C4,5 Programs for Machine Learning, Morgan Kaufmann Publishers, San Mateo (CA), 1993.

Refenes A-P. N., Zapranis A.D., Neural Model Identification, Variable Selection and Model Adequacy, *Journal of Forecasting*, 18, 1999, pp. 299-332.

Ripley B.D., Statistical Ideas for Selecting Network Architectures, *Neural Networks : Artificial Intelligence and Industrial Applications*, eds., B. Kappen and S. Gielen, London : Springer, 1995.

Ripley B.D., Neural Networks and related methods for classification (with discussion), *Journal of Royal Statistical Society Series*, B56, 1994, pp. 409-456.

Rissanen J., Modeling by shortest data description, *Automatica 14*, 1978, pp. 465-471.

Setiono R. and Liu H., Improving backpropagation learning with feature selection. *Journal of Applied Intelligence,* Vol. 6, No. 2, April 1996, pages 129-140.

Setiono R. and Liu H., Neural-network feature selector, *IEEE Transactions on Neural Networks,* Vol. 8, No. 3, May 1997, pp. 654-662.

Van De Laar P., Heskes T. and Gielen S., Partial Retraining : A New Approach to Input Relevance Determination, *International Journal of Neural Systems*, Vol. 9, No. 1, 1999, pp. 75-85.

Wang Z., Di Massimo C., Tham M.T. and Morris A.J., A Procedure for Determining the Topology of Multilayer Feedforward Neural Networks, *Neural Networks*, Vol. 7, 1994, pp. 291-300.